

# Decision Support System Architecture, Hardware, and Operating System Platforms

*Radiant Victor Imbar*

*Staf Pengajar Jurusan S-1 Sistem Informasi*

*Fakultas Teknologi Informasi*

*Universitas Kristen Maranatha*

*Jl.Prof. Drg. Suria Sumantri No. 65, BANDUNG 40164*

Email : [radiant.vi@eng.maranatha.edu](mailto:radiant.vi@eng.maranatha.edu)

## **Abstrak**

*DSS (Decision Support System) / Sistem Pendukung Keputusan adalah sistem informasi berbasis komputer yang tujuan utamanya adalah menyediakan informasi yang bisa menjadi dasar untuk pengambilan keputusan. Seperti sebuah perangkat lunak komputer berada dalam suatu lingkungan terintegrasi antara perangkat keras dengan sistem operasinya. Begitu pula dalam perencanaan DSS, bagaimana membangun DSS yang dapat membantu pengambilan keputusan. Selain itu DSS memerlukan arsitektur komputer yang tepat dalam pengaplikasiannya, meliputi perangkat keras dengan sistem operasi yang mendukung, memilih kombinasi yang tepat dan, atau dengan kata lain, arsitektur komputer yang tepat dapat membuat DSS berjalan dengan efektif dan efisien dan demikian pula sebaliknya.*

*Kata kunci : Komputer, DSS, Arsitektur.*

### **1. Definition of DSS and History of DSS.**

A DSS can be described as a computer-based interactive human-computer decision-making system that:

1. supports decision makers rather than replaces them.
2. utilizes data and models.
3. solves problems with varying degrees of structure.
4. focuses on effectiveness rather than efficiency in decision processes (facilitating decision processes).

DSS evolved early in the era of distributed computing. The history of such systems begins in about 1965. At about this time, the development of the [IBM System 360](#) and other more powerful mainframe systems made it more practical and cost-effective to develop Management Information Systems (MIS) in large companies (cf., Davis, 1974). MIS focused on providing managers with structured, periodic reports. Much of the information was from accounting and transaction systems.

By the late 1970s, a number of researchers and companies had developed interactive information systems that used data and models to help managers analyze semi-structured problems. These diverse systems were called Decision Support Systems. From those early days, it was recognized that DSS could be designed to support decision-makers at any level in an organization. DSS could support operations, financial management and strategic decision-making. DSS could use spatial data in a system like Geodata Analysis and Display System (GADS) (cf., Grace, 1976), structured multidimensional data and unstructured documents (cf., Swanson and Culnan, 1978). A variety of models were used in DSS including optimization and simulation. Also, statistical packages were recognized as tools for building DSS. Artificial Intelligence researchers began work on management and business expert systems in the early 1980s.

In the middle and late 1980s, Executive Information Systems (EIS) evolved from single user model-driven Decision Support systems and improved relational database products. The first EIS used pre-defined information screens and were maintained by analysts for senior executives.

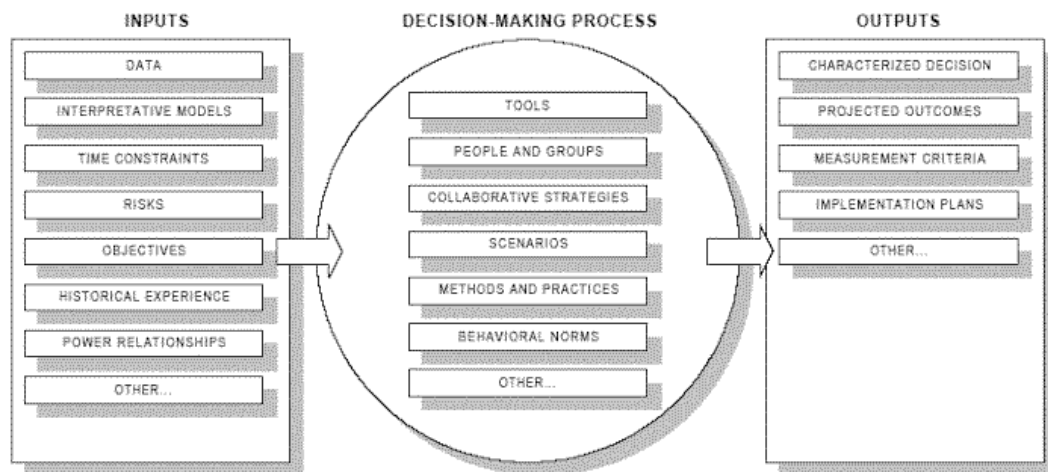
Beginning in about 1990, data warehousing and On-Line Analytical Processing (OLAP) began broadening the realm of EIS and defined a broader category of Data-Driven DSS (cf., Dhar and Stein, 1997). Nigel Pendse (1997) claims the first Executive Information System product was Pilot Software's Command Center. He notes both multidimensional analysis and OLAP had origins in the APL programming language and in systems like Express and Comshare's System W. Nigel Pendse of the [OLAPReport.com](http://OLAPReport.com) has written and updates a much more detailed history of the origins of OLAP products.

A DSS can take many different forms. Minimally we can say that a DSS is a system for making decisions. A decision is a choice between alternatives based on estimates of the values of those alternatives. Supporting a decision means supporting this choice by supporting the estimation, the evaluation and/or the comparison and choice. In practice references to DSS are usually references to computer applications that perform such a supporting role.

Who is the decision-maker? What kinds of data serve as inputs to the decision-making process? What does the decision-making process itself look like? What kinds of risks and constraints are associated with the decision-making process? How is the output of the decision-making process - a decision - evaluated, implemented and tracked?

Ultimately, no matter where we ground ourselves in a discussion of decision support systems, we have to develop and employ a model of decision-making: the set of activities that DSS environments support. The key elements of this model are fairly common, and include:

- a **decision-maker**: an individual or group charged with making a particular decision
- a set of **inputs to the decision-making process**: data, numerical or qualitative models for interpreting that data, historical experience with similar data sets or similar decision-making situations, and various kinds of cultural and psychological norms and constraints associated with decision-making
- the **decision-making process itself**: a set of steps, more or less well-understood, for transforming the inputs into outputs in the form of decisions,
- a set of **outputs from the decision-making process**, including the decisions themselves and (ideally) a set of criteria for evaluating decisions produced by the process against the set of needs, problems or objectives that occasioned the decision-making activity in the first place.



*Figure 1. A Prototypic Decision-Making Model (Partial)*

How do DSS environments support decision-making?

DSS environments support the generic decision-making model above in a number of ways:

- In decision preparation, DSS environments provide data required as input to the decision-making process. This is interestingly enough, about all most data mart and data warehousing environments do today.
- In decision structuring, DSS environments provide tools and models for arranging the inputs in ways that make sense to frame the decision. These tools and models are not pivot tables and other

aspects of data presentation found in query tools. They are actual decision making tools, like fault tree analysis, Bayesian logic and model-based decision-making based on things like neural networks.

- In context development, DSS environments again provide tools, and provide the mechanisms for capturing information about a decision's constituencies (who's affected by this decision), outcomes and their probabilities, and other elements of the larger decision making context.
- In decision-making, DSS environments may automate all or part of the decision-making process and offer evaluations on the optimal decision. Expert systems and artificial intelligence environments purport to do this, but they work only in very limited cases, because of some fundamental flaws in the technology (namely, their inability to deal with non-binary, or fuzzy, choices, like "it's more likely that we'll lose market share than win it," which is a rule that no traditional AI-based system can code).
- In decision propagation, DSS environments take the information gathered about constituencies and dependencies and outcomes and drive elements of the decision into those constituencies for action.
- In decision management, DSS environments inspect outcomes days, weeks and months after decisions to see if (a) the decision was implemented/propagated and (b) if the effects of the decision are as expected.

## 2. Defining The DSS Architecture.

An important issue to consider before planning individual systems is developing an overall enterprise information systems architecture. The architecture of an information system refers to the way its pieces are laid out, what types of tasks are allocated to each piece, how the pieces interact with each other, and how they interact with the outside world. Despite the unique nature of most DSS applications, several considerations that parallel the development of any other large-scale software application must be made. One of the most important considerations is the degree to which the proposed DSS conforms to and integrates with the existing enterprise information system architecture. The architecture of an information system refers to the manner in which the various pieces of the system are laid out with respect to location, connectivity, hierarchy, and internal and external interactions.

When we speak of an information systems architecture, we are not referring to the exact make and model of computer, disk drive, or monitor in the system. Rather, we are focused on three specific higher level issues:

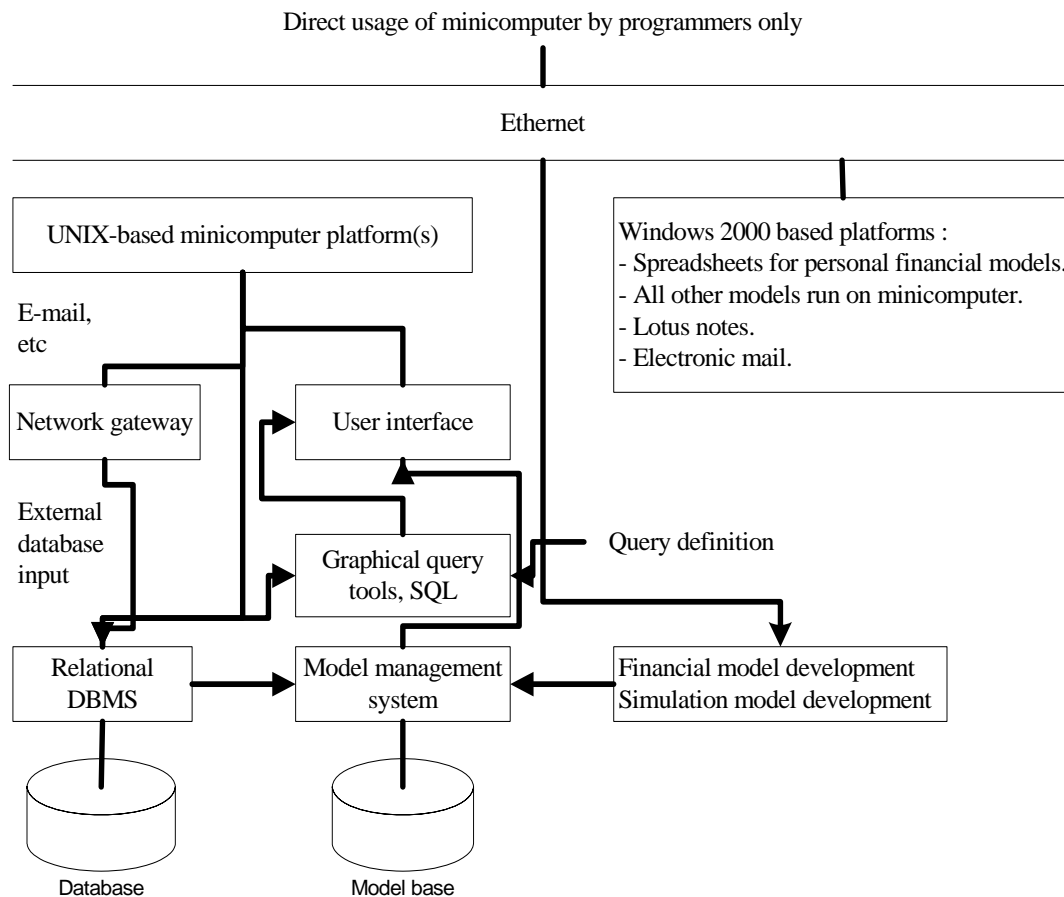
- **Interoperability**  
DSS interoperability controls to degree to which information can be delivered to the exact locus of a decision or point of user in an effective and efficient manner. It ensures that the DSS under design

can access and deliver its information to the appropriate end users at whatever location or time period they may need it.

- **Compatibility**  
Compatibility of systems, so that resources can be shared easily and leveraged across the organization.
- **Scalability**  
Expandability of systems, so that limited single-function components do not create bottlenecks that obstruct the growth of the organization.

The overall architecture of a DSS should be laid out and understood before specific hardware and software selection decisions are made. The nature of this architecture depends on the DSS. To lay out a DSS architecture must consider the spectrum of DSS that the organization will use :

- Strategic, tactical (management control), and operational decisions.
- Unstructured, semistructured, and structured decisions.
- All level of management and knowledge workers in the organization.
- All major functional, product or line of business, and geographic divisions of the organization.



*Figure 2. Specific DSS Architecture*

### 2.1. DSS and Client/Server Computing

Many organizations deal with the aforementioned disadvantages by providing individual users, via their desktop computers, with access to data on the central computer. The desktop computers then handle the computations and other processes of a DSS. These processes usually include supporting the usual elements of today's easy to use graphical user interfaces, such as windows, pull-down menus, and the use of a mouse or other pointing device. This approach is referred to as client/server computing. The system storing the database is called a server. The server system acts as a data repository. The client system, typically on the user's desktop, runs the application using data from the server. The result is a close matching of each partner's capabilities to its role in the overall system.

The advantages of using a client/server computing are client/server computing allows each part of an overall application to run in a hardware/software environment optimized for that part and that part alone, whether it demands a large database, high speed computing. This

optimization tend to minimize the overall cost of the system. Another advantage is because the client/server approach is increasingly popular, many applications and software development tools are designed to run in a client/server environment. Organizations can often reduce development time and cost by using this.

The disadvantages of using a client/server system are the complexities of application development and management are much greater in a client/server system than they are on a single computer, security also becomes a major concern. Security features, perhaps including firewalls to limit access from outside the organization.

## **2.2. The Internet and Client/Server Computing in DSS**

The Internet is a giant worldwide information source. The World Wide Web, especially, provides easy access to information on a wealth of topics. Much of this information can be useful for decision making. It is possible to take this same technology and this same interface and use them to access a corporate DSS. The Web architecture, the HyperText Markup Language (HTML) for developing Web pages, the Java language that most browsers can interpret, the JavaScript language for commands added to HTML, and the standards that Web browsers must follow define, in effect, a platform for applications. DSS can use this platform so that any user with a Web browser can access those DSS easily. The advantages are it is accessible from anywhere in the industrialized world and the web can be accessed via any type of hardware. The disadvantages is web access can be slow because a lot of graphics added to web pages. The hardware and communication links needed to provide fast access may be more expensive than local area network

## **2.3. DSS Using Shared Data on a Separate System**

It is not necessary for the server in a client/server system to be the central corporate computer that stores the live, operational database. It is often a good idea to extract meaningful decision support data from operational database and load it into a different computer.

The linked-system approach is effective when the application allows for a decision support database that is separate from the firm's transaction processing database. As you know, such a database is often called a data warehouse.

Because corporate data warehouses tend to be large, the servers required are correspondingly large. Multiprocessors, which harness the power of two to several dozen processors to a single overall task, are useful here because they increase the computing power of a system beyond that of a single processor without requiring the expensive technologies used in mainframes or supercomputers.

A data warehouse is a read-only database. Its users cannot update the "live" organizational information. That limits the use of a data warehouse to situations where decision makers access corporate information but do not change it. If an application running in this environment must update the main database, it must be done in some other way. One approach is to have DSS computer mimic a transaction processing user and submit transactions.

The Advantages of using Shared Data on a Separate System are DSS hardware need not to be shared with other applications, so DSS response time is preserved no matter what is going on elsewhere in the organization and the DSS hardware can be optimized for that purpose and that purpose alone. The disadvantages are the need to transfer data between two systems and users who access both systems may need two types of terminals or may have to follow complex network logon procedures to access the correct one. Switching from one to another may be time consuming.

#### **2.4. DSS on a Stand-Alone System**

DSS that do not access a large central database are candidates for stand-alone systems. A stand-alone DSS can be run on a computer dedicated to the DSS task or on a multiple-user computer used in time-sharing mode. Being "stand-alone" is a matter of degree. The absolutely, totally, 100 percent stand-alone system is almost as rare as the unicorn. Even users of nominally stand-alone system exchange files via universal serial bus (USB) or send e-mail via a dial-up modem link. What the term really means here is that the operation of the DSS doesn't depend on a regular connection to another computer. Such a connection probably does exist and can be used for DSS-related purposes, such as sending its results to a colleague for review and comment. The advantages of using a stand alone system are the system can be totally optimized for DSS and the complexity of sharing a resource with other users is avoided. The disadvantages of using a stand alone system are any data the system requires must be provided by its users and it may be difficult to integrate a stand alone system with corporate applications at a later date.

### **3. Choosing a DSS Hardware Environment**

Here is a list of questions that can ask to help make the choice among the previously listed options. As a group, though, their answers will help point out the right DSS hardware approach :

1. Are there any corporate policies that must follow ? If there are, they may narrow the choice by mandating one option, or by eliminating some options.
2. How large and widespread will the DSS user community be ? If it is large and widespread, will its members use exactly the same application, variations on one application, or different applications ? When they use the same application, do they use it



independently, or must their usage be coordinated through a shared database or in any other way ? The more similarity, the more sharing, the more should look toward shared systems and LAN with high-end servers. The less sharing and similarity, the more power probably belongs on individual desktops.

3. Are most of the prospective users already using a particular system ? If so, see if it can be used as is or with modifications such as modest upgrade or interconnection via a LAN.
4. Is there a corporate mainframe with sufficient capacity, or to which sufficient capacity can be added at reasonable cost ?
5. Is another powerful server system available, linked to that mainframe or not, with sufficient capacity ?
6. Do prospective users already have microcomputers or workstations that can handle the application ?
7. If new systems are required, will the existing central system be able to share data with them ? to shoulder an application jointly with them ?
8. Do the necessary development tools exist for any of those systems ? Are they already within the organization , available from the system vendor, or available from third parties ?
9. With which of these systems and tools, if any, are the prospective DSS developers already familiar ?
10. Does the application require access to a database ? If so, does that database already exist ? If not, is it to be derived from corporate data or from a separate source ? If from corporate data, how up to date should the database be for the application to function ?
11. Does the application's use of the corporate database require only the ability to read the data, or must it also be able to update the data ?
12. How much processing power does the application require ? How much data storage capacity ?
13. Are prospective users capable of performing (and willing to perform) basic system administration tasks, such as installing software and backing up data files ?

## Summary

DSS are information systems whose primary purpose is to provide knowledge workers with information. Common characteristics of most or all DSS include their use by managers and other knowledge workers, their use of a database and their use of models. DSS are generally used when a computer cannot be programmed to make a decision for all cases. They support, but do not replace, human decision makers.

Before embarking on the development of any major DSS, we should have a clear idea of both its architecture and the overall DSS architecture of the

organization. Knowing these will help to communicate the DSS vision to management and will help to plan systems that will continue to meet user needs in the future. DSS can run on a variety of different platforms. Each type has both advantages and disadvantages, which often make one better than another in a specific situation. An existing multiuser computer may be suitable when it has adequate capacity, prospective DSS users are already using it, and rapid access to central corporate data is required.

It is not necessary to provide decision support data directly from the live database. Instead, the necessary data can be consolidated into a specialized decision support database, often called a data warehouse. While this is not absolutely necessary, data warehouse are often housed on separate systems from the operational database. In other situations, model-oriented or process-oriented DSS often do not need a database at all. In these cases a stand-alone DSS, usually running on the user's desktop system with no active connections to the outside world, may be the best solution.

#### **BIBLIOGRAPHY**

Mallach, Efrem G (2000). *Decision Support and Data Warehouse System*: McGraw-Hill.

Stair, Ralph M., and George W. Reynolds (1997). *Principles of Information Systems: A Managerial Approach*: Cambridge.

Martin, E. Wainwright, Daniel W. DeHayes, Jeffrey A. Hoffer, and William C. Perkins (1991). *Managing Information Technology: What Managers Need to Know* : Macmillan, New York.

Demarest, Mark., Technology and Policy in Decision Support Systems, <http://dssresources.com/papers/features/demarest05/demarest07082005.html>. Tgl Akses 10 November 2006.

[http://en.wikipedia.org/wiki/Decision\\_support\\_system](http://en.wikipedia.org/wiki/Decision_support_system). Access date 10 November 2006.

[http://elearning.dmst.aueb.gr/mis/Cases/Nederlandse\\_Spoorweg/Cases/Training\\_Files/iebmddssrwwweb.pdf](http://elearning.dmst.aueb.gr/mis/Cases/Nederlandse_Spoorweg/Cases/Training_Files/iebmddssrwwweb.pdf). Access date 10 November 2006